Received 28 April 2023
Accepted 8 August 2023

DOI: 10.48308/CMCMA.2.1.1
AMS Subject Classification: 49M15; 65H05; 65F15

Transforming Ostrowski's method into a derivative-free method and its dynamics

Vali Torkashvand[a]

The current research develops a derivative-free family without memory methods. The proposed method consisting of two steps and one parameter for solving nonlinear equations is brought forward. The basin of attraction of the proposed methods has investigated using different weight functions. Numerical examples are experimented with to check the performance of the proposed schemes. Furthermore, the theoretical order of convergence is confirmed on the experiment work.

Keywords: Iterative method; Convergence order; Basin of attraction; Nonlinear equation.

1. Introduction

Most of the Mathematical problems that arise in science and engineering are very hard and sometime impossible to solve exactly. Therefore, it is indispensable to calculate approximate solutions based on numerical methods. The celebrated Newton's method can define as $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$, is one of the oldest and the most applicable methods in the literature. This method has locally quadratically convergence for the simple roots and per iteration requires one evaluation of the function and its first derivative. About two centuries later, in 1960, the first optimal two-point method was constructed by Ostrowski [13]

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \; k = 0, 1, 2, \cdots, \\ x_{k+1} = y_k - \frac{f(y_k)(y_k - x_k)}{2f(y_k) - f(x_k)}. \end{cases} \tag{1}$$

This method can be rewritten as follows:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \; k = 0, 1, 2, \cdots, \\ x_{k+1} = y_k - \frac{f(y_k)}{f'(x_k)} \frac{f(x_k)}{f(x_k) - 2f(y_k)}. \end{cases} \tag{2}$$

Also, this family of two-point methods requires three function evaluations and has the order of convergence four. Therefore, this family is of optimal order and supports the Kung-Traub conjecture [10]. The efficiency of the new method is measured by the concept of efficiency index. Commonly, the efficiency of an iterative method is measured by the efficiency index defined by Ostrowski in [13] as $\sqrt[d]{p}$, where p is the order of convergence and d is the number of functional evaluations per step. Accordingly, the efficiency index of Newton's method and Ostrowski's method are respectively: $\sqrt{2} \approx 1.41$ and $\sqrt[3]{4} \approx 1.58$. But these methods have a major weakness, one has to calculate the derivative of $f(x)$ at each approximation. In this work, we will turn Ostrowski's method into a Steffensen-type. In this way, the problem of computing the derivative of the function is solved. We consider approximating the derivative function by the divided difference method. The construction of the proposed class has based on the weight function approach. We have described the structure of the modified Ostrowski's methods two-step without memory in Section two. The numerical study presented in Section 3 confirms the theoretical results. In Section 4, the dynamical properties of the proposed methods along with their illustrative basins of attraction and weight functions have been displayed with detailed analyses and comments. Finally, we give the concluding remarks.

[a] *Department of Mathematics, Farhangian University, Tehran, Iran. Email: torkashvand1978@gmail.com.*
[*] *Correspondence to: V. Torkashvand.*

2. Construction of Without Memory Method

By looking at relation (2), it can be seen that this method uses the derivative of the function in the first and second steps and this reveals that the two-point family of methods (2) reaches at least fourth order of convergence by using only three functional evaluations (i.e., $f(x_n)$, $f(y_n)$, and $f'(x_n)$) per full iteration. To derive new methods, we approximate $f'(x_n)$ given in one-step (2) as follows:

$$w_k = x_k + \beta f(x_k),\ f'(x_k) \approx f[w_k, x_k] = \frac{f(w_k) - f(x_k)}{w_k - x_k}. \tag{3}$$

In following, the derivative $f'(x_n)$ in the second step will be approximated by

$$\frac{f[y_k, w_k]}{h(t_k)} \tag{4}$$

where $h(t_k)$ is a differentiable function that depends real variable $t_k = \dfrac{f(y_k)}{f(x_k)}$. Therefore, we start from the scheme (2), the approximations (3), (4) and state the following two-point method

$$\begin{cases} w_k = x_k + \beta f(x_k),\ y_k = x_k - \frac{f(x_k)}{f[w_k, x_k]},\ k = 0, 1, 2, \cdots, \\ x_{k+1} = y_k - H(t_k)\frac{f(x_k)}{f(x_k) - 2f(y_k)}\frac{f(y_k)}{f[y_k, w_k]}. \end{cases} \tag{5}$$

Theorem (1) illustrates that under what conditions on weight function, convergence order of two-step family (5) will arrive at the optimal level four.

Theorem 1 *Let $H, f : D \subset \mathbb{R} \to \mathbb{R}$ have a single root $x^* \in D$, for an open interval D. If the initial point x_0 is sufficiently close to x^*, then the sequence x_m generated by any method of the family (5) converges to x^*. If H is any function with $H(0) = 1,\ H'(0) = -1,\ H''(0) < \infty$ and $\beta \neq 0$ then the methods defined by (5) have convergence order at least 4.*

Proof. By using Taylor's expansion of $f(x)$ about x^* and taking into account that $f(x^*) = 0$, $e_k = x_k - x^*$, also c_k, for $k \geq 2$, are defined by $c_k = \frac{f^{(k)}(x^*)}{k! f'(x^*)}$. We obtain

$$f(x_k) = f'(x^*)(e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4 + O(e_k^5)). \tag{6}$$

Then, computing $e_{k,w} = w_k - x^*$, we attain $w_k = x_k + \beta f(x_k)$

$$e_{k,w} = e_k + e_k \beta f'(x^*)(1 + e_k(c_2 + e_k(c_3 + e_k c_4))) + O(e_k^5), \tag{7}$$

and

$$\begin{aligned} f(w_k) =& f'(x^*)(e_k + e_k\beta f'(x^*)(1 + e_k(c_2 + e_k(c_3 + e_k c_4))) + c_2(e_k + e_k\beta f'(x^*) \\ & (1 + e_k(c_2 + e_k(c_3 + e_k c_4))))^2 + c_3(e_k + e_k\beta f'(x^*)(1 + e_k(c_2 + e_k \\ & (c_3 + e_k c_4))))^3 + c_4(e_k + e_k\beta f'(x^*)(1 + e_k(c_2 + e_k(c_3 + e_k c_4))))^4). \end{aligned} \tag{8}$$

Considering $f[x, y] = \frac{f(x) - f(y)}{x - y}$ is Newton's first order divided difference. we get

$$\begin{aligned} f[x_k, w_k] =& -1/(e_k\beta f'(x^*)(1 + e_k(c_2 + e_k(c_3 + e_k c_4))))^{-1}(e_k f'(x^*)(1 + e_k(c_2 + e_k(c_3 \\ & + e_k c_4))) - f'(x^*)(e_k + e_k\beta f'(x^*)(1 + e_k(c_2 + e_k(c_3 + e_k c_4))) + c_2(e_k + e_k\beta \\ & f'(x^*)(1 + e_k(c_2 + e_k(c_3 + e_k c_4))))^2 + c_3(e_k + e_k\beta f'(x^*)(1 + e_k(c_2 + e_k(c_3 \\ & + e_k c_4))))^3 + c_4(e_k + e_k\beta f'(x^*)(1 + e_k(c_2 + e_k(c_3 + e_k c_4))))^4)). \end{aligned} \tag{9}$$

So that

$$\begin{aligned} y_k =& x^* + (1 + \beta f'(x^*))e_k^2 + (-(2 + \beta f'(x^*)(2 + \beta f'(x^*))c_2^2) + (1 + \beta f'(x^*))(2 + \beta f'(x^*)) \\ & c_3 e_k^3 + ((4 + \beta f'(x^*)(5 + \beta f'(x^*)(3 + \beta f'(x^*))))c_2^3 - (7 + \beta f'(x^*)(10 + \beta f'(x^*) \\ & (7 + 2\beta f'(x^*))))c_2 c_3 + (1 + \beta f'(x^*)(3 + \beta f'(x^*)(3 + \beta f'(x^*)))c_4)e_k^4 + O(e_k^5). \end{aligned} \tag{10}$$

The expansion of $f(y_k)$ about x^* is given as

$$\begin{aligned} f(y_k) =& f'(x^*)(1+\beta f'(x^*))c_2e_k^2 + f'(x^*)(-(2+\beta f'(x^*)(2+\beta f'(x^*))c_2^2 + (1+\beta f'(x^*)) \\ & (2+\beta f'(x^*))c_3e_k^3 + f'(x^*)((5+\beta f'(x^*)(7+\beta f'(x^*)(4+\beta f'(x^*))))c_2^3 - \\ & (7+\beta f'(x^*)(10+\beta f'(x^*)(7+2\beta f'(x^*))))c_2c_3 + (1+\beta f'(x^*) \\ & (3+\beta f'(x^*)(3+\beta f'(x^*)))c_4)e_k^4 + O(e_k^5). \end{aligned} \tag{11}$$

From (6) and (11), we now have

$$\begin{aligned} \frac{f(x_k)}{f(x_k)-2f(y_k)} =& 1+2(1+f'(x^*))c_2e_k + 2((-1+\beta f'(x^*)(1+f'(x^*)))c_2^2(1+f'(x^*)) \\ & (2+f'(x^*))c_3)e_k^2 + 2(f'(x^*)(-1+\beta f'(x^*))(2+\beta f'(x^*))c_2^3 + 2(-1+\beta f'(x^*)) \\ & (1+\beta f'(x^*))(3+f'(x^*))c_2c_3 + (1+f'(x^*))(3+f'(x^*)(3+f'(x^*)))c_4)e_k^3 + O(e_k^4). \end{aligned} \tag{12}$$

Using (8) and (11), we obtain

$$\begin{aligned} \frac{f(y_k)}{f[w_k,y_k]} =& (1+f'(x^*))c_2e_k^2 + (-(3+2\beta f'(x^*)(2+f'(x^*)))c_2^2(1+f'(x^*))(2+f'(x^*)) \\ & c_3)e_k^3 + ((7+f'(x^*)(11+\beta f'(x^*)(8+\beta f'(x^*))))c_2^3 - 2(5+\beta f'(x^*) \\ & (9+\beta f'(x^*)(7+2f'(x^*))))c_2c_3 + (1+f'(x^*))(3+f'(x^*)(3+f'(x^*)))c_4)e_k^4 + O(e_k^5). \end{aligned} \tag{13}$$

Using the Taylor expansion $H(t_k)$ we have

$$H(t_k) = H(\frac{f(y_k)}{f(x_k)}) = H(0) + H'(0)*(\frac{f(y_k)}{f(x_k)}) + H''(0)*\frac{(\frac{f(y_k)}{f(x_k)})^2}{2}. \tag{14}$$

Now with using relations $h0 = H(0)$, $h1 = H'(0)$, $h2 = H''(0)$, and from (14), we get,

$$\begin{aligned} H(t_k) =& h0 + h1(1+\beta f'(x^*))c_2e_k + (\frac{1}{2}(h2((1+\beta f'(x^*))^2 - 2h1(3+\beta f'(x^*)(3+\beta f'(x^*)))) \\ & c_2^2 + h1(1+\beta f'(x^*))(2+\beta f'(x^*))c_3e_k^2 + ((-h2(1+\beta f'(x^*)(3+\beta f'(x^*)(3+\beta f'(x^*))) \\ & + h1(2+\beta f'(x^*))(4+\beta f'(x^*)(3+\beta f'(x^*))))c_2^3 + h2(1+\beta f'(x^*))^2(2+\beta f'(x^*)) \\ & - 2h1(5+\beta f'(x^*)(7+\beta f'(x^*)(4+\beta f'(x^*))))c_2c_3 + h1(1+\beta f'(x^*))(3+\beta f'(x^*) \\ & (3+\beta f'(x^*)))c_4)e_k^3 + O(e_k^4). \end{aligned} \tag{15}$$

Thus, substituting (10), (12), (13) and (15) in (5), we get

$$\begin{aligned} x_{k+1} - x^* = y_k - x^* - H(t_k)&\frac{f(x_k)}{f(x_k)-2f(y_k)}\frac{f(y_k)}{f[y_k,w_k]} \\ =& -(-1+h0)(1+\beta f'(x^*))c_2e_k^2 + ((-2+h0-h1(1+\beta f'(x^*))^2 - \beta f'(x^*) \\ & (2+\beta f'(x^*)))c_2^2 - (-1+h0)(1+\beta f'(x^*))(2+\beta f'(x^*))c_3)e_k^3(\beta f'(x^*))^2 \\ & + (\frac{1}{2}(8-2h0+8h1-h2+f'(x^*)(10+6h0+14h1-10h2)(6+8h1-3h2) \\ & (-\beta f'(x^*))^3(-2+2h0-2h1+h2)c_2^3 - (7-h0+4h1+2\beta f'(x^*) \\ & (5+h0+5h1) + (\beta f'(x^*))^2(7+2h0+8h1) + 2(\beta f'(x^*))^3(1+h1)c_2c_3 \\ & - (-1+h0)(1+\beta f'(x^*))(3+\beta f'(x^*)(3+\beta f'(x^*))))c_4)e_k^4 + O(e_k^5). \end{aligned} \tag{16}$$

By putting $h0 = 1$, $h1 = -1$, the final error expression is given by:

$$e_{k+1} = \frac{-1}{2}((1+\beta f'(x^*))^2c_2)((-2+h2+f'(x^*)\beta(2+h2)c_2^2 + 2c_3))e_k^4 + O(e_k^5). \tag{17}$$

Hence, the fourth-order convergence is established. □

Some of the functions that Satisfy to Theorem 1 are as follows:

$$\begin{cases} H_1(t) = 1 - t,\ H_2(t) = \frac{1}{1+t},\ H_3(t) = (1 - \frac{t}{2})^2,\ H_4(t) = e^{-t},\ H_5(t) = \frac{1+2t}{1+3t}, \\ H_6(t) = \cos(t) - \sin(t),\ H_7(t) = Arccos(t),\ H_8(t) = \frac{t^2+1}{1+t},\ H_9(t) = e^t - 2t. \end{cases} \tag{18}$$

Using iterative methods, many efficient multipoint iterative methods have been proposed for solving nonlinear equations, see [5, 7, 16, 17, 20].

3. Numerical results

The principal purpose of numerical examples is to verify the validity of the theoretical developments through a variety of test examples using high accuracy computations by use of Mathematica program. All computations were done by using Mathematica 11.

We have used many "*SetAccuracy*, *MantissaExponent*, *FindRoot*, *andWorkingPrecision*" commands in the computer programs of this article. We explain each of the following:

1. When *SetAccuracy* is used to increase the accuracy of a number, the number is padded with zeros. The zeros are taken to be in base 2. In base 10, the additional digits are usually not zeros. *SetAccuracy* returns an arbitrary-precision number even if the number of significant digits obtained will be less than MachinePrecision. When *expr* contains machine-precision numbers, $SetAccuracy[expr, a]$ can give results that differ from one computer system to another. *SetAccuracy* will first expose any hidden extra digits in the internal binary representation of a number, and, only after these are exhausted, add trailing zeros.
2. $MantissaExponent[x]$ gives a list containing the mantissa and exponent of a number x. $MantissaExponent[x, b]$ gives the base-b mantissa and exponent of x.
3. $FindRoot[f, \{x, x_0\}]$ searches for a numerical root of f, starting from the point $x = x_0$. $FindRoot[lhs == rhs, \{x, x_0\}]$ searches for a numerical solution to the equation $lhs == rhs$. $FindRoot[\{f_1, f_2, \cdots\}, \{\{x, x_0\}, \{y, y_0\}, \cdots\}]$ searches for a simultaneous numerical root of all the f_i.
4. *WorkingPrecision* is an option for various numerical operations that specifies how many digits of precision should be maintained in internal computations [6].

In tables one through five, the abbreviations *Div*, *TNE* and *Iter* are used as follows:
TNE: Total Number of Evaluations required for a method to do the specified iterations.
Div: The corresponding iterative method is divergent for the initial guess.
Iter: Number of repetitions
The errors $|x_k - \alpha|$ of approximations to the corresponding zeros of test functions $f_i(x)$, $i = 1, 2 \cdots, 10$.
The computational order of convergence r_c [14] computed by the expressions (if they are stable)

$$r_c = \frac{\log |f(x_n)/f(x_{n-1})|}{\log |f(x_{n-1})/f(x_{n-2})|}, \tag{19}$$

We shall check the effectiveness of the new without memory methods. We employ the presented methods (5) denoted by TM4, (for different values of β) to solve some nonlinear equations. We compared our methods and some known methods as follows: Jarratt's method (JM) [9]:

$$\begin{cases} w_k = \frac{f(x_k)}{f'(x_k)},\ y_k = x_k - \frac{1}{2}\frac{f(x_k)}{f'(x_k)},\ k = 0, 1, 2, \cdots, \\ x_{k+1} = y_k + \frac{f(x_k)}{f'(x_k) - 3f'(x_k - \frac{2}{3}w_k)}. \end{cases} \tag{20}$$

Kung-Traub'method(KTM) [10]:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)},\ k = 0, 1, 2, \cdots, \\ x_{k+1} = y_k - \frac{f(x_k)f(y_k)}{(f(x_k) - f(y_k))^2}\frac{f(x_k)}{f'(x_k)}, \end{cases} \tag{21}$$

Maheshwari's method (MM) [11]:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)},\ k = 0, 1, 2, \cdots, \\ x_{k+1} = x_k + \frac{1}{f'(x_k)}\left(\frac{f(x_k)^2}{f(y_k) - f(x_k)} - \frac{f(y_k)^2}{f(x_k)}\right). \end{cases} \tag{22}$$

Ostrwoski's method (OM) (1) [13]. And , Traub's method (TM) [18]:

$$\begin{cases} \lambda_k = \frac{-1}{f[x_k, x_{k-1}]},\ k = 1, 2, \ldots, \\ w_k = x_k + \lambda_k f(x_k),\ x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k]},\ k = 0, 1, \ldots. \end{cases} \tag{23}$$

The numerical values in Tables 1-3 validate that the presented scheme TM4 performs better, not only for the absolute error in the root and the absolute value of the function as compared to without memory method. In Tables 1-3 we show the numerical results obtained by applying the different methods with memory for approximating the solution of $f_i(x) = 0,\ i = 1, 2, 3$. It should note that the condition for the convergence of repetitive methods is to select the appropriate initial conjecture root of the nonlinear equation. One can see more about this in Reference [15].
$f_1(x) = x^5 + x^4 + 4x^2 - 15,\ x^* = 1.34,\ x_0 = 1.1$, [19]
$f_2(x) = \cos(x^2 - 1) - x\log(x^2 - \pi) + 1,\ x^* = \sqrt{1+\pi},\ x_0 = 2$, [8]
$f_3(x) = \sqrt{x^4+8}\ sin(\frac{\pi}{x^2+2}) + \frac{x^3}{x^4+1} - \sqrt{6} + \frac{8}{17},\ x^* = -2,\ x_0 = -2.3$ [15]

Table 1. Numerical results.

functions		OM [13]	JM [9]	KTM [10]	MM [11]
f_1	$\lvert x_{n+1} - x_n\rvert$	9.19935×10^{-171}	3.75861×10^{-43}	5.39338×10^{-31}	1.08801×10^{-18}
	$\lvert f(x_{n+1})\rvert$	1.47556×10^{-43}	4.04445×10^{-169}	4.4005×10^{-120}	2.1393×10^{-70}
	Iter	3	4	4	4
	r_c	4.00	4.00	4.00	3.99
f_2	$\lvert x_{n+1} - x_n\rvert$	1.06006×10^{-81}	4.00433×10^{-79}	1.90086×10^{-69}	6.23543×10^{-62}
	$\lvert f(x_{n+1})\rvert$	5.11850×10^{-323}	1.26383×10^{-312}	1.84157×10^{-273}	5.15151×10^{-243}
	Iter	3	4	4	4
	r_c	4.00	4.00	4.00	4.00
f_3	$\lvert x_{n+1} - x_n\rvert$	7.74534×10^{-70}	1.08145×10^{-52}	2.21403×10^{-36}	2.771×10^{-29}
	$\lvert f(x_{n+1})\rvert$	2.22649×10^{-279}	2.03962×10^{-210}	2.30758×10^{-144}	1.62563×10^{-115}
	Iter	3	4	4	4
	r_c	4.00	4.00	4.00	4.00

Table 2. Numerical results.

functions		TM4(5), $H_1(t)$	TM4(5), $H_2(t)$	TM4(5), $H_3(t)$	TM4(5), $H_4(t)$	TM4(5), $H_5(t)$
	β_0	$\beta_0 = 0.1$	$\beta_0 = 0.01$	$\beta_0 = -0.01$	$\beta_0 = 0.01$	$\beta_0 = 0.01$
f_1	$\lvert x_{n+1} - x_n\rvert$	1.08041×10^{-10}	1.72662×10^{-15}	2.51132×10^{-13}	7.74905×10^{-8}	1.34597×10^{-13}
	$\lvert f(x_{n+1})\rvert$	4.37754×10^{-37}	1.14922×10^{-57}	4.55033×10^{-50}	1.86151×10^{-27}	1.10928×10^{-49}
	Iter	3	3	3	3	3
	r_c	3.99	3.99	3.99	4.00	4.00
	β_0	$\beta_0 = 1$	$\beta_0 = -0.1$	$\beta_0 = 1$	$\beta_0 =$	$\beta_0 = 1$
f_2	$\lvert x_{n+1} - x_n\rvert$	6.22292×10^{-90}	3.79045×10^{-14}	5.325×10^{-10}	2.99541×10^{-7}	7.07869×10^{-6}
	$\lvert f(x_{n+1})\rvert$	9.08601×10^{-360}	1.56633×10^{-51}	4.50474×10^{-33}	5.29199×10^{-22}	4.08921×10^{-16}
	Iter	3	3	3	3	3
	r_c	3.99	3.99	3.99	3.99	4.02
	β_0	$\beta_0 = 0.1$	$\beta_0 = 0.01$	$\beta_0 = 1$	$\beta_0 = 1$	$\beta_0 = 0.1$
f_3	$\lvert x_{n+1} - x_n\rvert$	1.30669×10^{-13}	1.93275×10^{-10}	1.45948×10^{-22}	1.28419×10^{-22}	12.55487×10^{-9}
	$\lvert f(x_{n+1})\rvert$	3.18881×10^{-54}	1.13357×10^{-40}	2.48677×10^{-90}	1.33337×10^{-90}	8.76226×10^{-36}
	Iter	3	3	3	3	3
	r_c	3.99	4.00	4.00	4.00	4.00

In the continuation of this work, in the next section, the dynamic behavior of the proposed method for different values of β and the choice of weighted functions in Equation (18) will be examined.

4. Attraction basins of fourth-order derivative-free methods

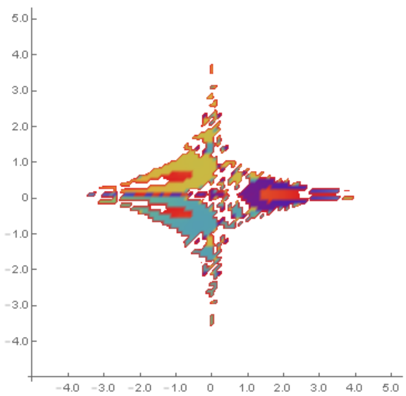
In this section, to analyze the dynamic behavior of the proposed method, selecting the appropriate value of the parameter β and also selecting the weight function with the maximum absorption region of three polynomial functions have been used. From the dynamical point of view, we take a 500×500 grid of the square $D = [-5, 5] \times [-5, 5] \in \mathbf{C}$. Several iterative root-nding methods have compared from a dynamical point of view by Babajee et al. [1], Chicharro et al. [2], Chun et al. [3], Cordero et al. [4], Geum et al. [8], Moccari-Lotfi [12]. We have studied the dynamic behavior of the proposed methods by using the following two functions:

$$f(z) = z^2 - 1,\ f(z) = z^3 - 1 \tag{24}$$

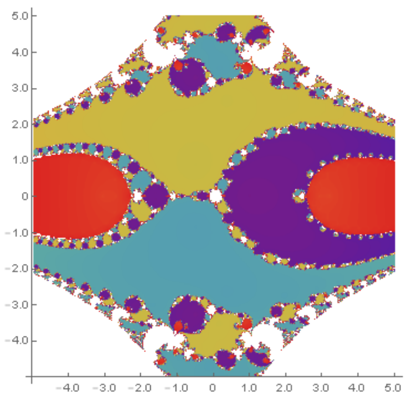
Table 3. Numerical results.

functions		TM4(5), $H_6(t)$	TM4(5), $H_7(t)$	TM4(5), $H_8(t)$	TM4(5), $H_9(t)$
f_1	β_0	$\beta_0 = 0.01$	$\beta_0 = 0.1$	$\beta_0 = 0.01$	$\beta_0 = 0.01$
	$\lvert x_{n+1} - x_n \rvert$	1.91966×10^{-7}	0.00091×10^{0}	1.24753×10^{-6}	9.33817×10^{-9}
	$\lvert f(x_{n+1}) \rvert$	8.51553×10^{-26}	0.00009×10^{0}	4.79146×10^{-11}	3.92571×10^{-31}
	Iter	3	3	3	3
	r_c	3.98	2.01	2.00	4.00
f_2	β_0	$\beta_0 = 1$	$\beta_0 = -1$	$\beta_0 = -1$	$\beta_0 = -0.1$
	$\lvert x_{n+1} - x_n \rvert$	2.92092×10^{-21}	0.00976×10^{0}	0.00976×10^{0}	6.66773×10^{-15}
	$\lvert f(x_{n+1}) \rvert$	1.95807×10^{-78}	0.00957×10^{0}	0.00957×10^{0}	8.49775×10^{-55}
	Iter	3	3	3	3
	r_c	4.00	3.42	3.42	3.99
f_3	β_0	$\beta_0 = 0.01$	$\beta_0 = -3$	$\beta_0 = -3$	$\beta_0 = 0.1$
	$\lvert x_{n+1} - x_n \rvert$	2.25856×10^{-11}	0.15816×10^{0}	0.15816×10^{0}	1.61482×10^{-13}
	$\lvert f(x_{n+1}) \rvert$	1.31937×10^{-44}	0.01699×10^{0}	0.01699×10^{0}	1.69108×10^{-53}
	Iter	3	3	3	3
	r_c	3.99	3.89	3.89	4.00

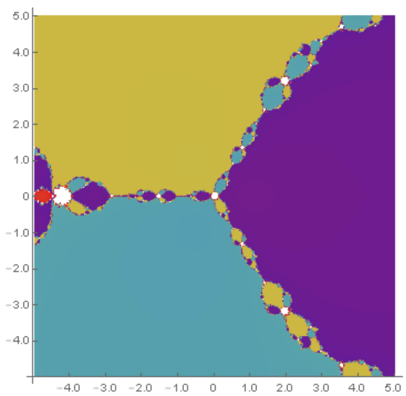
Figure 1 shows that the proposed method has the minimum the basins of attraction in $\beta = 1$ ((1a)) and the maximum the basins of attraction in $\beta = 0.01$ ((1c)).



(a) $H_1(t) = 1 - t$, $\beta = 1$



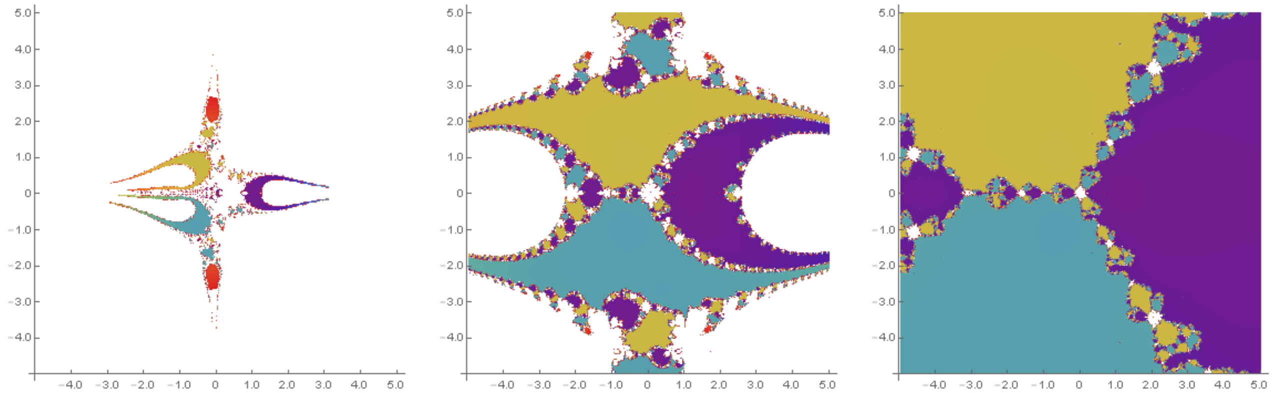
(b) $H_1(t) = 1 - t$, $\beta = 0.1$



(c) $H_1(t) = 1 - t$, $\beta = 0.01$

Figure 1. Method TM4 (5) for finding the roots of the polynomial $f(z) = z^3 - 1$

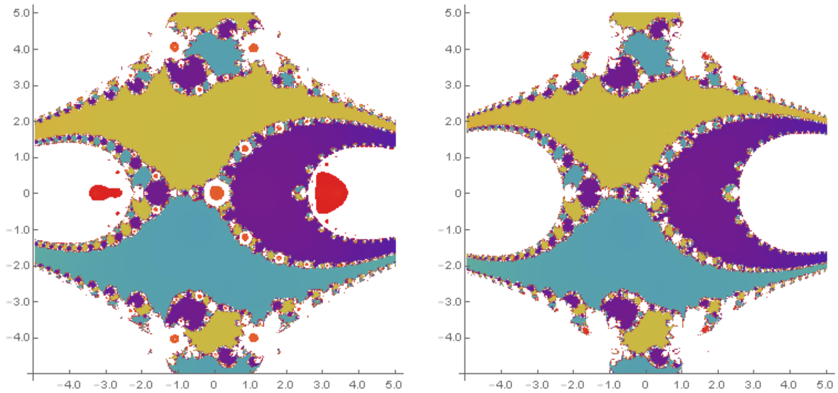
Figure 2 displays the basins of attraction of the proposed method by considering the weight function $H_2(t)$ with three values of β in three separate forms. From this figure, it results that the maximum area of absorption corresponds to $\beta = 0.01$ and also, the smallest-corresponds to $\beta = 1$.



(a) $H_2(t) = \frac{1}{1+t}$, $\beta = 1$ (b) $H_2(t) = \frac{1}{1+t}$, $\beta = 0.1$ (c) $H_2(t) = \frac{1}{1+t}$, $\beta = 0.01$

Figure 2. Method TM4 (5) for finding the roots of the polynomial $f(z) = z^3 - 1$

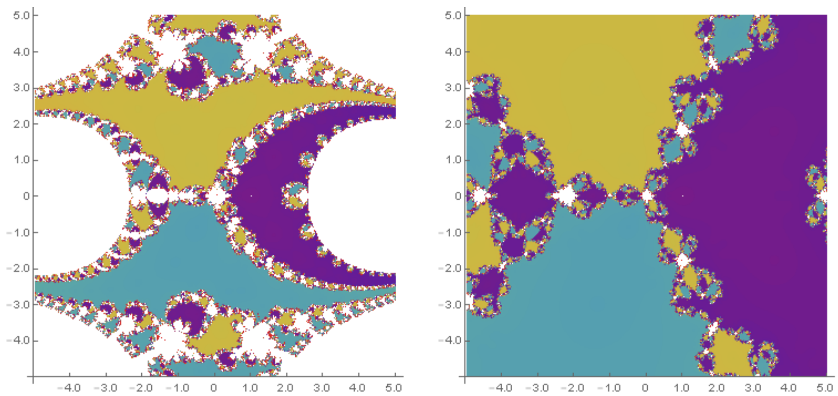
Figure 3 shows the the basins of attraction of the proposed method by considering the weight function $H_3(t)$ with two values of β in two separate forms. From this figure it follows that the maximum the basins of attraction corresponds to $\beta = 0.01$ and also the minimum corresponds to $\beta = 0.1$.



(a) $H_3(t) = (1 - \frac{t}{2})^2$, $\beta = 0.1$ (b) $H_3(t) = (1 - \frac{t}{2})^2$, $\beta = 0.01$

Figure 3. Method TM4 (5) for finding the roots of the polynomial $f(z) = z^3 - 1$

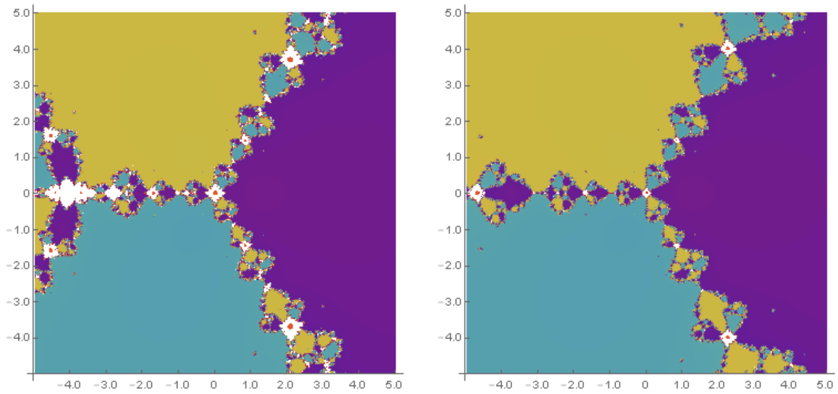
Figure 4 shows the polynomiographs of the proposed methods (5) for the cubic polynomial $p_1(z) = z^3 - 1$ with weight function $H_7(t)$ and $\beta = 0.1$, $\beta = 0.01$. Basins of attraction for TM4 (5) is illustrated in Figure 5. We have used the weight function $H_8(t)$



(a) $H_7(t) = \arccos(t)$, $\beta = 0.1$ (b) $H_7(t) = \arccos(t)$, $\beta = 0.01$

Figure 4. Method TM4 (5) for finding the roots of the polynomial $f(z) = z^3 - 1$

and the parameter β with values 0.1 and 0.01.



(a) $\beta = 0.01,\ H_8(t) = \frac{t^2+1}{1+t}$ (b) $\beta = 0.001,\ H_8(t) = \frac{t^2+1}{1+t}$

Figure 5. Method TM4 (5) for finding the roots of the polynomial $f(z) = z^3 - 1$

Figure 6 presents the polynomiographs of the suggested methods for the cubic polynomial $f(z) = z^3 - 1$ with weight function $H_5(t)$ and $\beta = 0.01$.

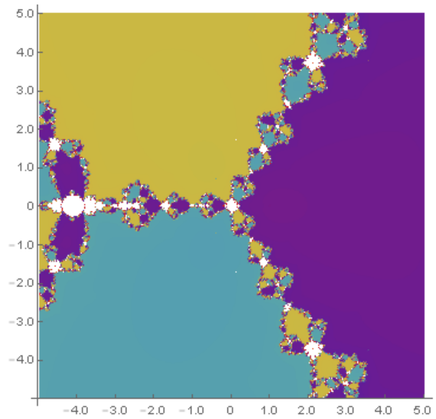
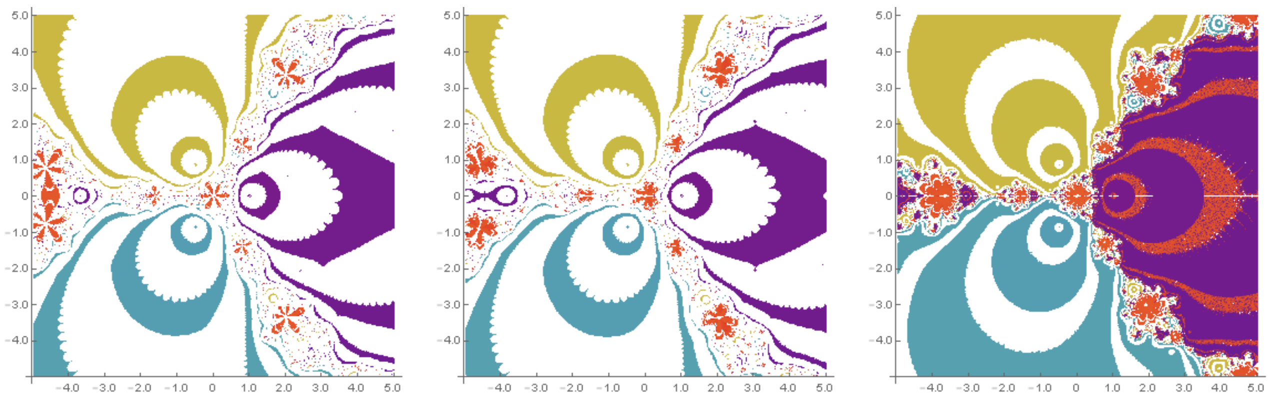


Figure 6. Method TM4 (5) for finding the roots of the polynomial $f(z) = z^3 - 1,$ with $\beta = 0.01,\ H_5(t) = \frac{1+2t}{1+3t}$

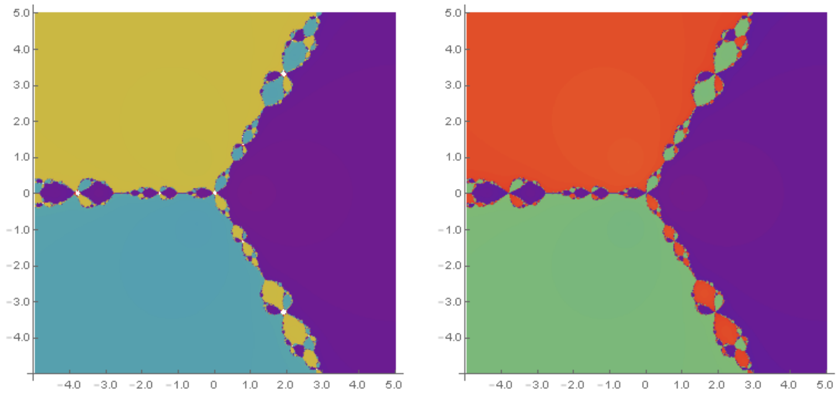
Figure 7 manifests the basins of attraction of the proposed method using three weight functions $H_4(t)$, $H_9(t)$, and $H_6(t)$ in Figures 7a, 7b, and 7c, respectively. Here the value of the free parameter $\beta = 0.01$ is recognized. The highest absorption domain among these three functions are related to function $H_6(t)$.



(a) $H_4(t)$, $\beta = 0.01$ (b) $H_9(t)$, $\beta = 0.01$ (c) $H_6(t)$, $\beta = 0.01$

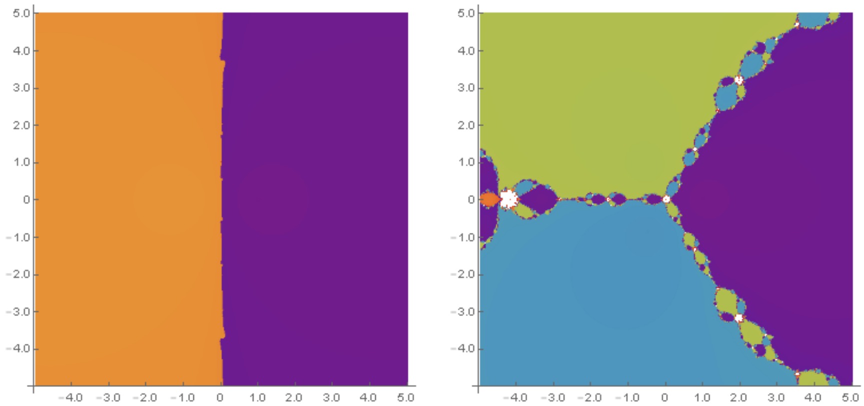
Figure 7. Method TM4 (5) for finding the roots of the polynomial $f(z) = z^3 - 1$

Figure 8 compares the basins of attraction of the proposed method using the weight function $H_1(t)$ and Ostrowski's method. Here the value of the free parameter $\beta = 0.001$ is considered. The absorption range of both-methods is almost the same. Besides, the proposed method does not use a function derivative to find the root of nonlinear equations.



(a) $\beta = 0.001,\ H_1(t) = 1 - t$ (b) $Ostrowski's\ Method$

Figure 8. Comparison basins of attraction of proposed methods TM4 (5) and Ostrowski's method for finding the roots of the polynomial $f(z) = z^3 - 1$



(a) $\beta = 0.01,\ H_1(t),\ f(z) = z^2 - 1$ (b) $H_1(t),\ f(z) = z^3 - 1,\ \beta = 0.01$

Figure 9. Comparison basins of attraction of proposed methods TM4 (5)

5. Conclusions

In this paper, we have used the idea of the weight function. Then, we have turned Ostrowski's method into an optimal-method. Numerical tests intend to verify the better performance of the proposed method over the others. According to the examples studied in Figures 1 to 9, we conclude that the weight function $H_1(t)$ and parameter $\beta = 0.001$ have the highest stability region and are competitive on Ostrowskis method.

Acknowledgments

The author wishes to state their sincere thanks to the referees for their very useful comments and suggestions for improving this work.

References

1. D. K. R. Babajee, A. Cordero, F. Soleymani, J. R. Torregrosa. On improved threestep schemes with high efficiency index and their dynamics. *Numerical Algorithms*, 65 (1): 153–169, 2014.
2. F. Chicharro, A. Cordero, J. M. Gutierrez, J. R. Torregrosa. Complex dynamics of derivative-free methods for nonlinear equations. *Applied Mathematics and Computation*, 219 (12): 7023–7035, 2013.

3. C. Chun, B. Neta, J. Kozdon, M. Scott. Choosing weight functions in iterative methods for simple roots. *Applied Mathematics and Computation*, 227: 788–800, 2014 .
4. A. Cordero, F. Soleymani, J .R. Torregrosa, S. Shateyi. Basins of attraction for various Steffensen-type methods. *Journal of Applied Mathematics*, 2014: 1–17, 2014.
5. A. Cordero, N. Garrido, J. R. Torregrosa, P. TriguerоNavarro. Memory in the iterative processes for nonlinear problems. *Mathematical Methods in the Applied Sciences*, 46 (4): 4145–4158, 2023.
6. E. Don, Schaum's Outline of Mathematica, McGraw-Hill Professional, New York, 2000.
7. R. Erfanifar, A class of efficient derivative free iterative method with and without memory for solving nonlinear equations. *Computational Mathematics and Computer Modeling with Applications (CMCMA)*, 1 (2): 20–26, 2022.
8. Y. H. Geum, Y. I. Kim, A. A. Magreñán. A biparametric extension of Kings fourth-order methods and their dynamics. *Applied Mathematics and Computation*, 282: 254–275, 2016.
9. P. Jarratt. Some Fourth order multipoint iterative methods for solving equations. *Mathematics of Computation*, 20: 434–437, 1966.
10. H. T. Kung, J. F. Traub. Optimal order of one-point and multipoint iteration. *J. Assoc. Comput. Mach*, 21 (4): 643–651, 1974.
11. A. K. Maheshwari. A fourth-order iterative method for solving nonlinear equations. *Applied Mathematics and Computation*, 211: 383–391, 2009.
12. M. Moccari, T. Lot. On a two-step optimal Steffensen-type method: Relaxed local and semi-local convergence analysis and dynamical stability. *Journal of Mathematical Analysis and Applications*, 468 (1): 240–269, 2018.
13. A. M. Ostrowski. Solution of equations and systems of equations, Academic press, New York, 1960.
14. M. S. Petković, B. Neta, L. D. Petković, J. Džunić. Multipoint Methods for Solving Nonlinear Equations, Elsevier, Amsterdam, 2013.
15. V. Torkashvand, T. Lotfi, M. A. Fariborzi Araghi. A new family of adaptive methods with memory for solving nonlinear equations. *Mathematical Sciences*, 13: 1–20, 2019.
16. V. Torkashvand. A two-step method adaptive with memory with eighth-order for solving nonlinear equations and its dynamic. *Computational Methods for Differential Equations*, 10 (4):1007–1026, 2022.
17. V. Torkashvand. A two-point eighth-order method based on the weight function for solving nonlinear equations. *Journal of Numerical Analysis and Approximation Theory*, 50 (1): 73–93, 2021.
18. J. F. Traub. Iterative Methods for the Solution of Equations, Prentice Hall, New York, USA, 1964.
19. M. Z. Ullah, V. Torkashvand, S. Shateyi, M. Asma. Using matrix eigenvalues to construct an iterative method with the highest possible efficiency index two. *Mathematics*, 10 (1370): 1–15, 2022.
20. F. Zafar, A. Cordero, J.R. Torregrosa. A family of optimal fourthorder methods for multiple roots of nonlinear equations. *Mathematical Methods in the Applied Sciences*, 43 (14): 7869–7884, 2020.

